Réalisation d'un feu tricolore avec Arduino UNO

Pour vous initier à la puissance de la carte programmable **UNO** de la famille **Arduino**, nous vous proposons dans cette activité de découverte de réaliser un feu tricolore simple à base de trois diodes LED et une carte Arduino UNO.

Pour rappel, une **carte Arduino UNO** est une carte électronique programmable développée et commercialisée par la société **Arduino** très populaire, que nous allons utiliser dans cette activité pour vous initier à la câbler, la programmer et à tester notre projet interactif d'un **feu tricolore**.

Schéma proposé d'un feu tricolore

Pour réaliser un feu tricolore à base de trois diodes LED de couleur ; vert, jaune et rouge, nous allons réaliser le schéma ci-dessous à l'aide d'une plaque d'essai :

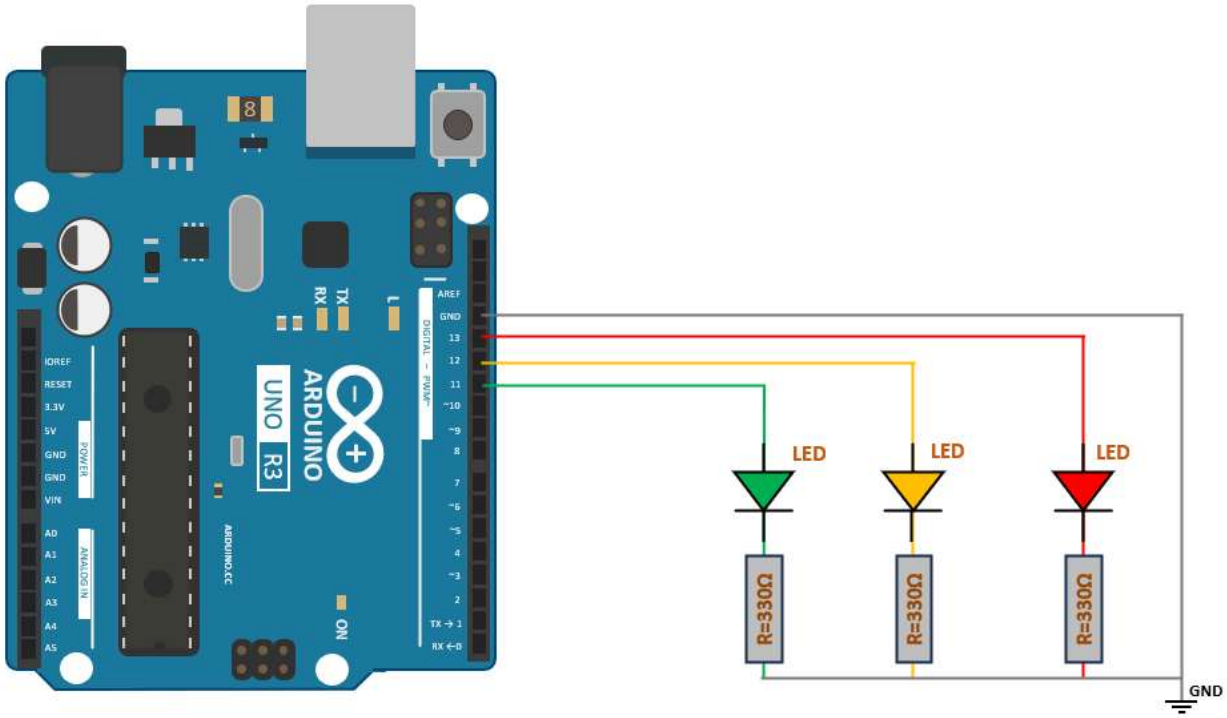


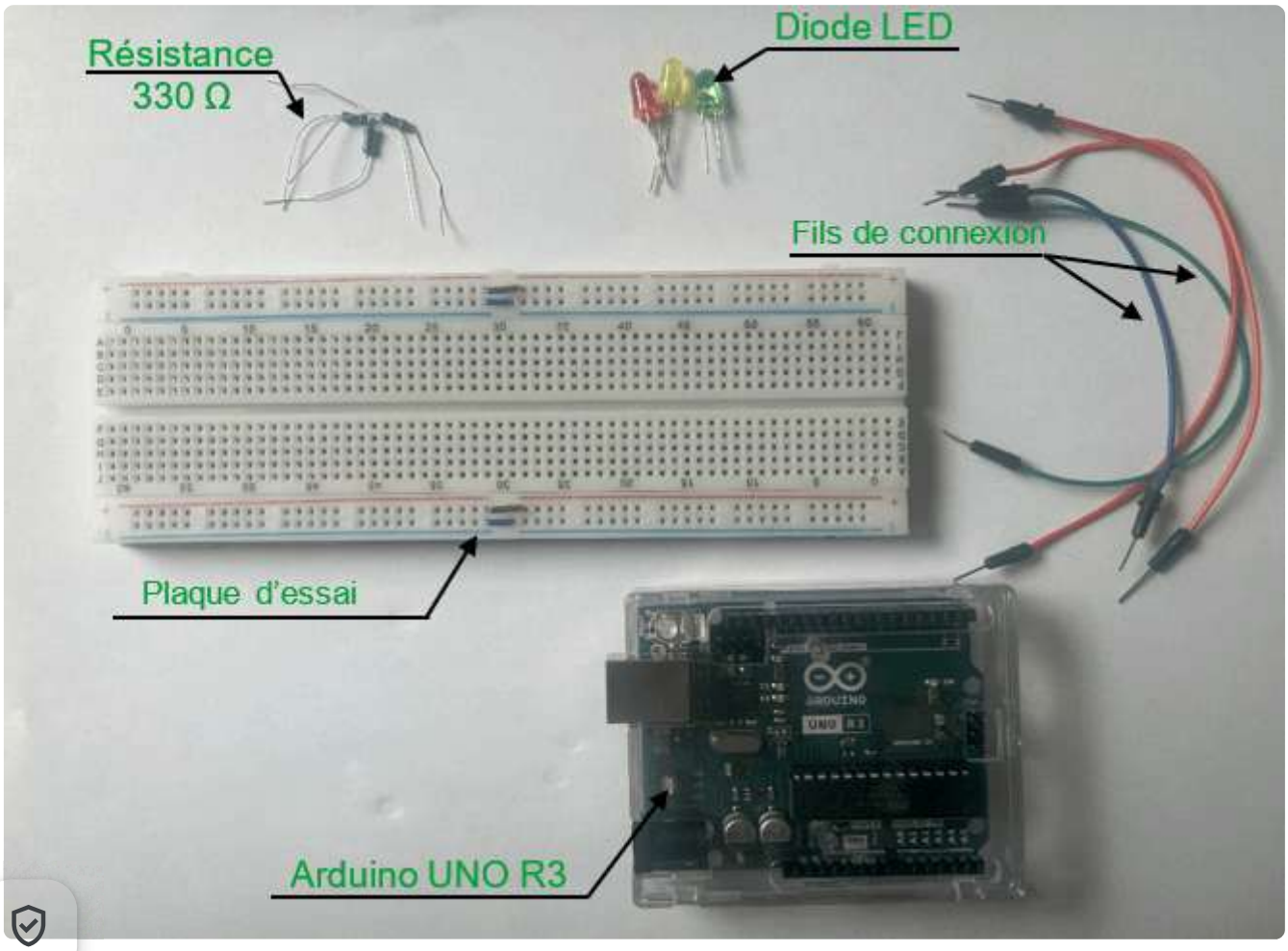
Schéma d'un feu tricolore à base d'une carte Arduino UNO

Comme nous pouvons le remarquer sur notre schéma, nous aurons besoin des composants suivants :

- Une plaque d'essai,

- Trois **résistances** d'une valeur chacune de **330 Ohm**,
- Des fils de connexion,
- Et une carte **Arduino UNO R3**

Ainsi, l'image ci-dessous vous détaille ces différents composants que nous avons préparer pour cette activité de découverte :



Par protection, nous avons mis une résistance d'une valeur de **330Ω** en série avec chaque **diode LED**.

Pour vérifier une résistance électrique de type résistance céramique exprimée en Ohm, nous pouvons utiliser la calculatrice proposée sur notre site electro-robot pour calculer les **codes couleur des résistances**.
Dans l'ordre, uné résistance d'une valeur de **330Ω** doit avoir les bandes de couleurs suivantes dans l'ordre de gauche à droite : **orange** pour la première bande, **orange** pour la deuxième bande, **noir** pour la troisième bande, **noir** pour la quatrième bande et la couleur **marron** pour la dernière.

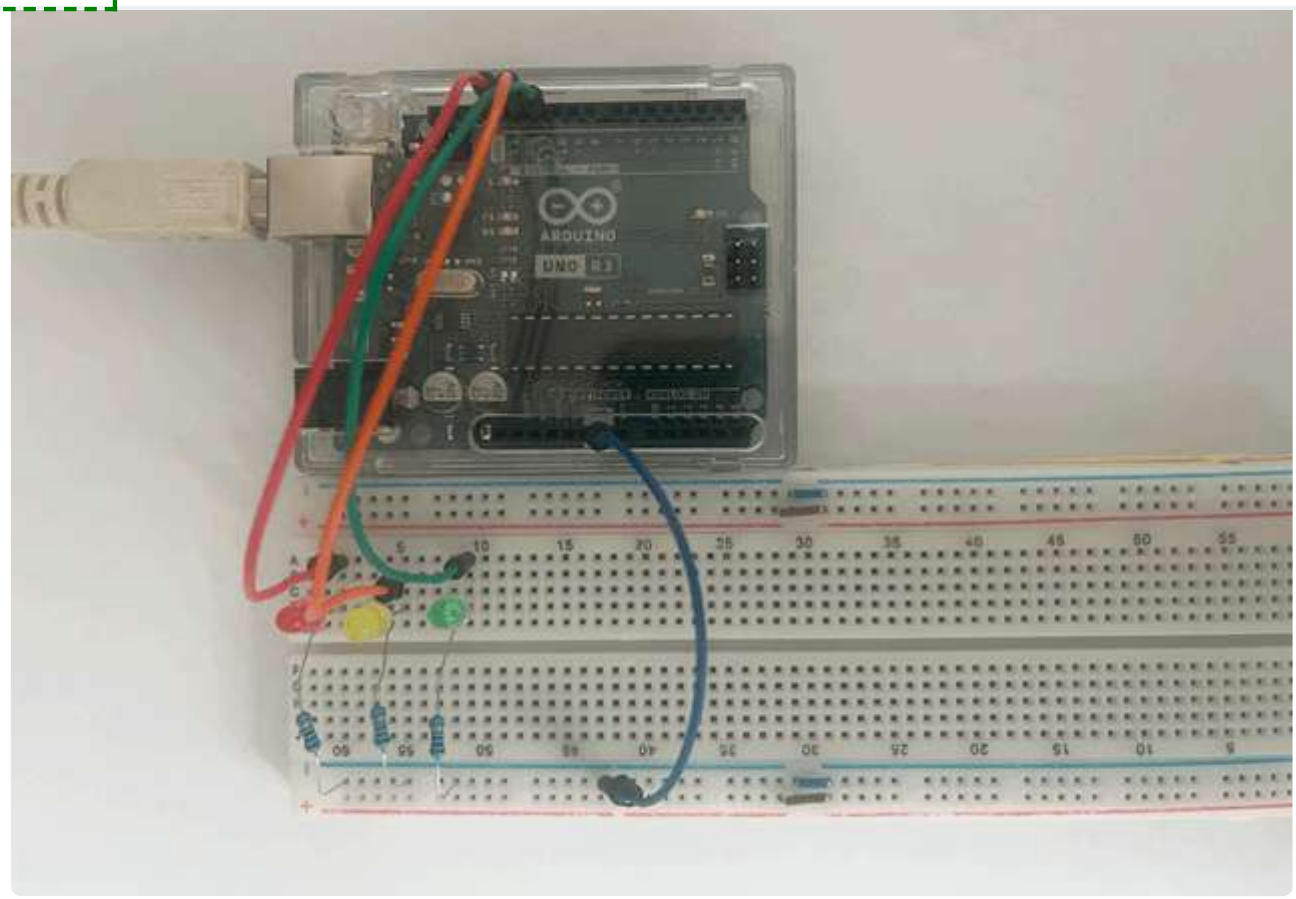
Câblage du Montage électrique et codage

Montage du schéma électrique proposé

Comme vous pouvez le constater dans le schéma fourni au début de cette activité, nous allons utiliser 3 pins de notre carte **Arduino UNO**, à citer:

- **Pin 13** : pour alimenter la lampe rouge, soit dans notre montage la diode LED rouge,
- **Pin 12** : pour alimenter la lampe jaune ou orange, soit dans notre montage la diode LED jaune,
- Et le **pin 11** : pour alimenter la lampe verte, soit dans notre montage la diode LED verte.

Une fois que nous avons réalisé le câblage de notre schéma proposé, notre montage devra ressembler à la photo ci-dessous :



Attention : Nous vous rappelons qu'il est recommandé de faire attention lorsque vous manipulez votre carte Arduino. Etant donné que vous n'avez pas encore acquis des compétences avancées en électronique, n'utilisez jamais une source d'énergie externe avec votre carte. Cela pourra vous endommager votre carte **Arduino UNO**.

Comme vous pouvez le constater, nous n'utilisons aucune source d'énergie externe dans notre montage et que:

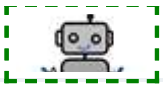
- Chaque fil représente une ligne pour alimenter une **diode LED** montée en série avec une résistance de protection,
- Et le fil bleu dans notre montage représente notre **GND.**

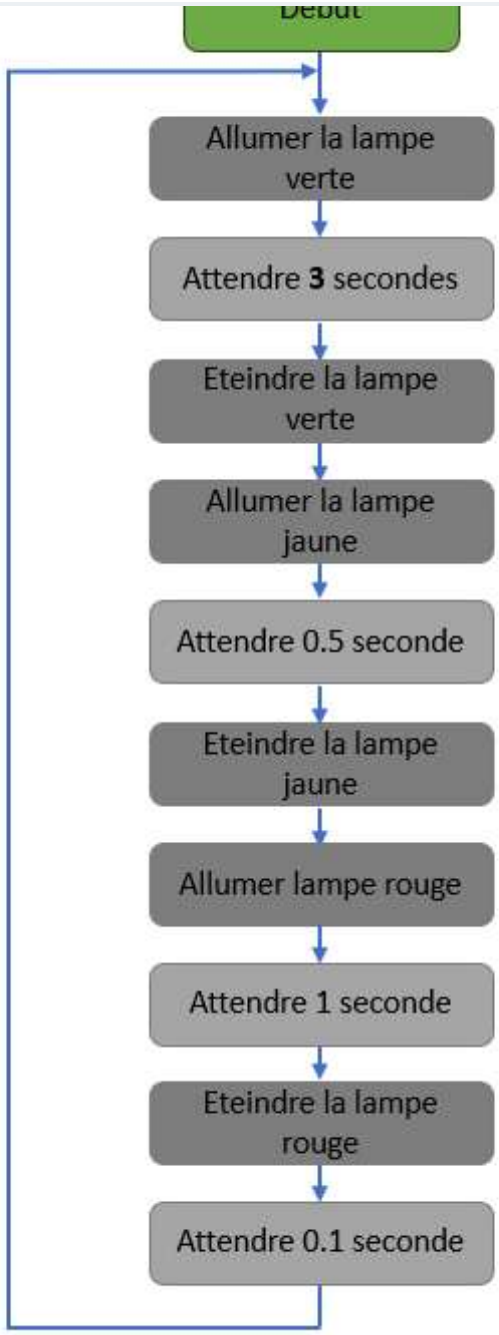
Codage, compilation et téléversement

Avant de procéder à l'écriture de notre code en **langue C++** sous **Arduino IDE**, nous allons définir un scénario de fonctionnement pour notre appliation **feu tricolore** tel que:

1. Allumer la lampe verte pendant 3 secondes,
2. Allumer la lampe jaune pendant 0.5 seconde,
3. Allumer la lampe rouge pendant 1 seconde,
4. Attendre 0.1 seconde avant de répéter le scénario → aller à l'étape 1.

Pour mieux comprendre ce mode de fonctionnement, nous vous proposons de vous réécrire notre algorithme sous forme de l'organigramme ci-dessous :





Ainsi, le code devient simple à écrire sous **Arduino IDE.** Au début, nous allons définir trois constantes pour désigner nos trois **diodes LED.** Ensuite, nous allons attribuer le mode **output** aux 3 pins 13, 12 et 11 pour les définir autant que des ports digitaux de sortie dans la fonction **setup().** Et finalement, nous pouvons écrire notre code dans la fonction **loop().**

Si nous décomposons notre organigramme, nous pouvons conclure que nous avons uniquement 3 **instructions** qui se répètent de base à écrire :

- **Allumer une diode LED** → le code source en C++ sous Arduino pour activer un port à la sortie est le suivant :

```
digitalWrite(ledVert, HIGH);
```

la sortie est le suivant :

```
digitalWrite(ledJaune, LOW);
```

- **Attendre n seconde(s)** → pour temporiser l'exécution de notre code source, le code source en C++ sous Arduino est le suivant :

```
delay(1000);
```

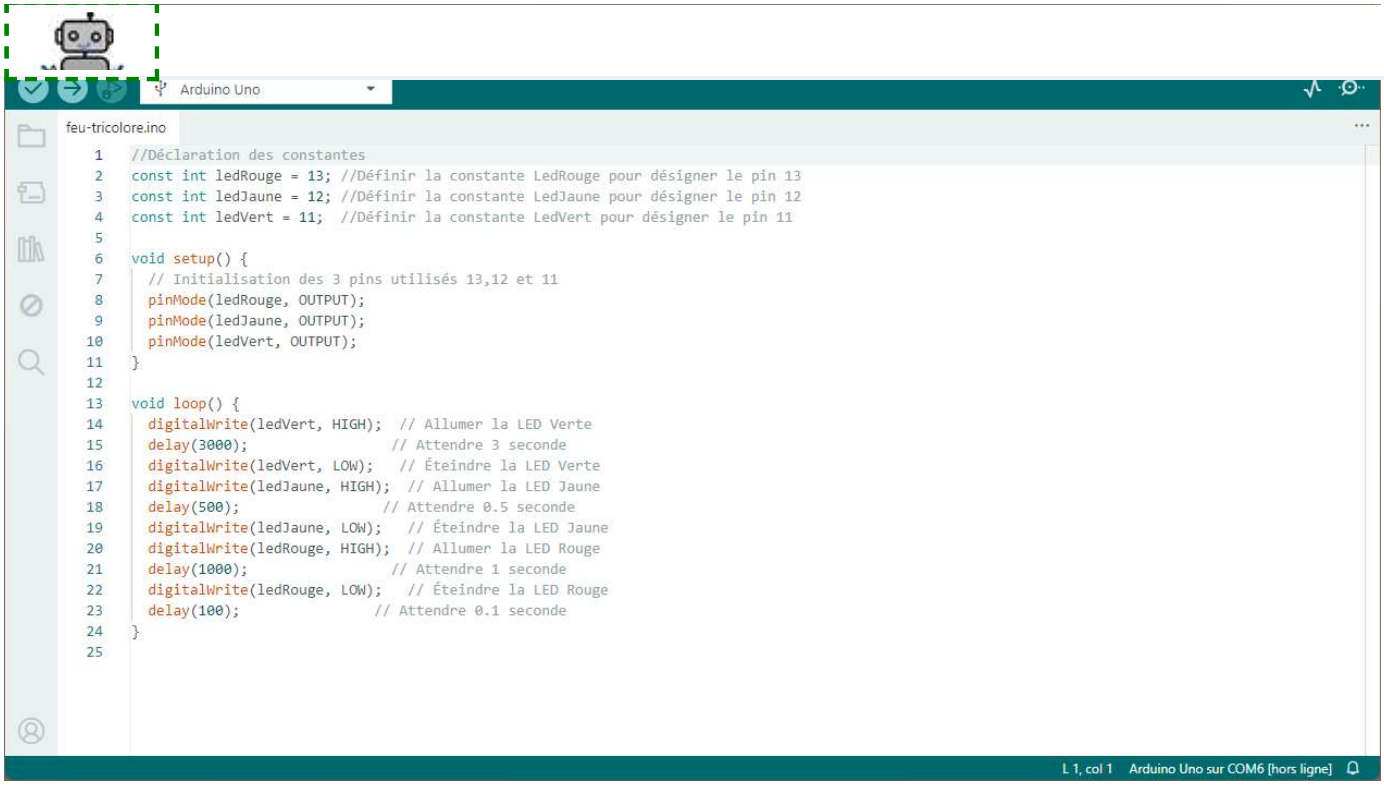
Maintenant, si nous écrivons notre code sous **Arduino IDE** en respectant le scénario de fonctionnement décrit ci-dessus, nous aurons le code complet suivant :

```
//Déclaration des constantes
const int ledRouge = 13; //Définir la constante LedRouge pour désigner le pin 13
const int ledJaune = 12; //Définir la constante LedJaune pour désigner le pin 12
const int ledVert = 11;  //Définir la constante LedVert pour désigner le pin 11

void setup() {
  // Initialisation des 3 pins utilisés 13,12 et 11
  pinMode(ledRouge, OUTPUT);
  pinMode(ledJaune, OUTPUT);
  pinMode(ledVert, OUTPUT);
}

void loop() {
  digitalWrite(ledVert, HIGH);  // Allumer la LED Verte
  delay(3000);                 // Attendre 3 seconde
  digitalWrite(ledVert, LOW);   // Éteindre la LED Verte
  digitalWrite(ledJaune, HIGH);  // Allumer la LED Jaune
  delay(500);                 // Attendre 0.5 seconde
  digitalWrite(ledJaune, LOW);   // Éteindre la LED Jaune
  digitalWrite(ledRouge, HIGH);  // Allumer la LED Rouge
  delay(1000);                 // Attendre 1 seconde
  digitalWrite(ledRouge, LOW);   // Éteindre la LED Rouge
  delay(100);                // Attendre 0.1 seconde
}
```

L'image ci-dessous nous montre notre code écrit sous **Arduino IDE** :


Arduino Uno
feu-tricolore.ino
1 //Déclaration des constantes
2 const int ledRouge = 13; //Définir la constante LedRouge pour désigner le pin 13
3 const int ledJaune = 12; //Définir la constante LedJaune pour désigner le pin 12
4 const int ledVert = 11; //Définir la constante LedVert pour désigner le pin 11
5
6 void setup() {
7 // Initialisation des 3 pins utilisés 13,12 et 11
8 pinMode(ledRouge, OUTPUT);
9 pinMode(ledJaune, OUTPUT);
10 pinMode(ledVert, OUTPUT);
11 }
12
13 void loop() {
14 digitalWrite(ledVert, HIGH); // Allumer la LED Verte
15 delay(3000); // Attendre 3 seconde
16 digitalWrite(ledVert, LOW); // Éteindre la LED Verte
17 digitalWrite(ledJaune, HIGH); // Allumer la LED Jaune
18 delay(500); // Attendre 0.5 seconde
19 digitalWrite(ledJaune, LOW); // Éteindre la LED Jaune
20 digitalWrite(ledRouge, HIGH); // Allumer la LED Rouge
21 delay(1000); // Attendre 1 seconde
22 digitalWrite(ledRouge, LOW); // Éteindre la LED Rouge
23 delay(100); // Attendre 0.1 seconde
24 }
25
L 1, col 1 Arduino Uno sur COM6 [hors ligne]

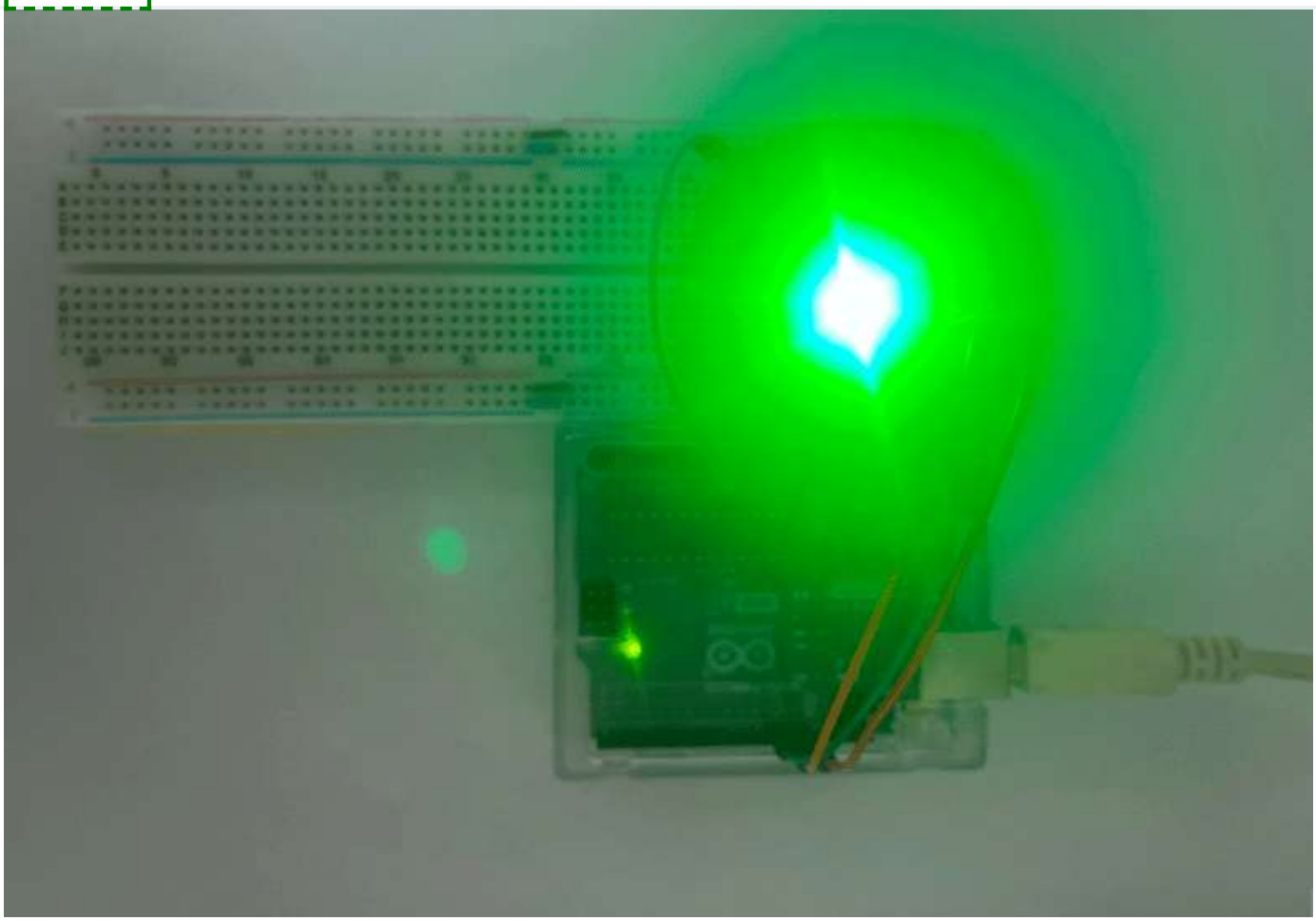
Une fois que vous avez terminé d'écrire votre code sous **Arduino IDE**, vous pouvez connecter votre carte Arduino UNO à votre ordinateur via un câble **USB-B** et puis lancer la compilation et le téléversement.

Si vous auriez une erreur lors de la compilation de votre code, nous vous recommandons de vérifier votre code source écrit sous **Arduino IDE**.
Et si vous auriez une erreur lors du téléversement du code compilé vers votre carte, nous vous recommandons de vérifier la configuration de votre port sous Arduino IDE. Pour cela, vous serez amené à vérifier cela dans le gestionnaire de cartes accessible sous **Arduino IDE** à partir du raccourci **Ctrl+Maj+B**

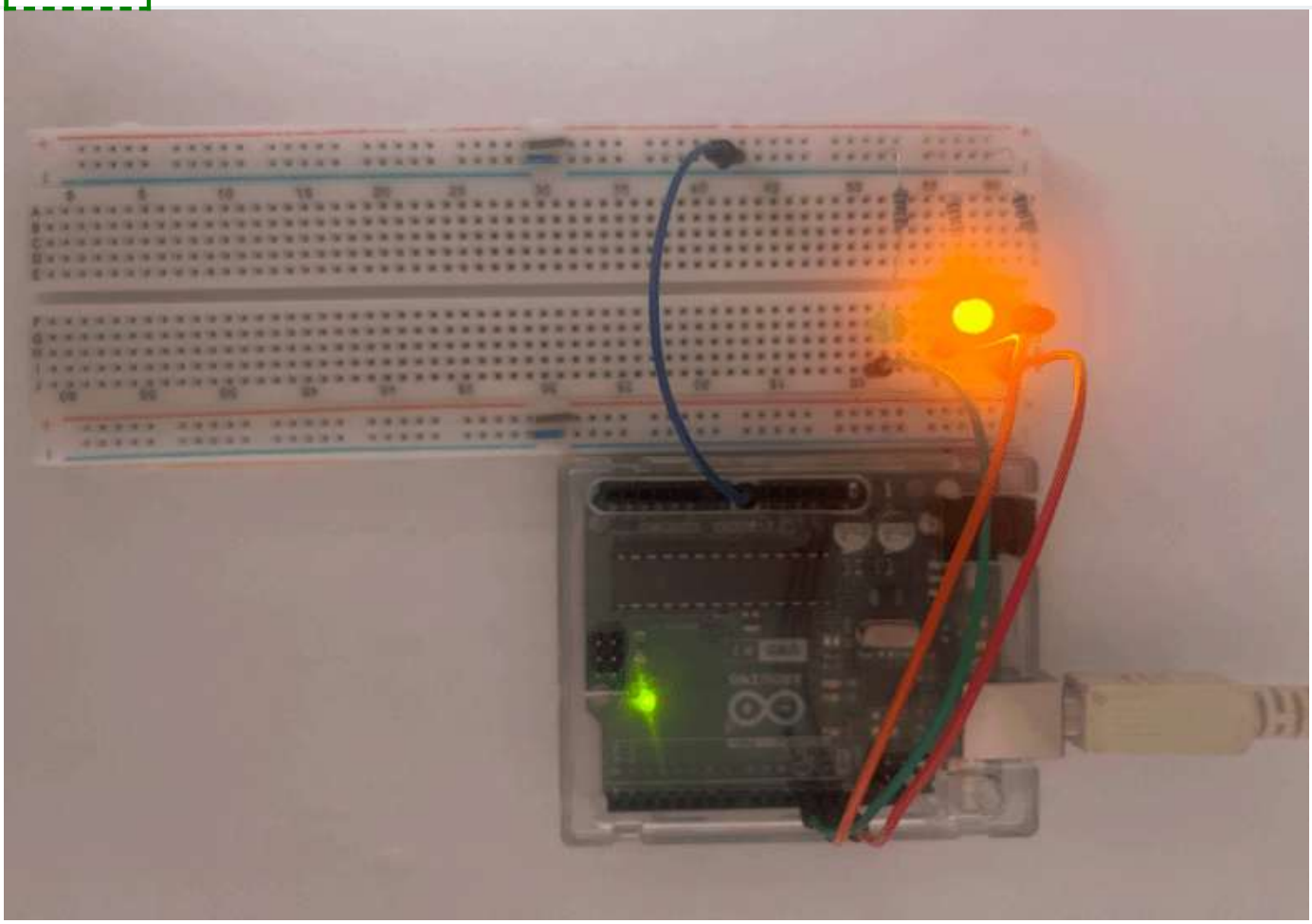
Exécution et conclusion

Enfin, après que nous avons tout câblé, codé et téléversé notre code vers notre carte, nous obtenons le résultat :

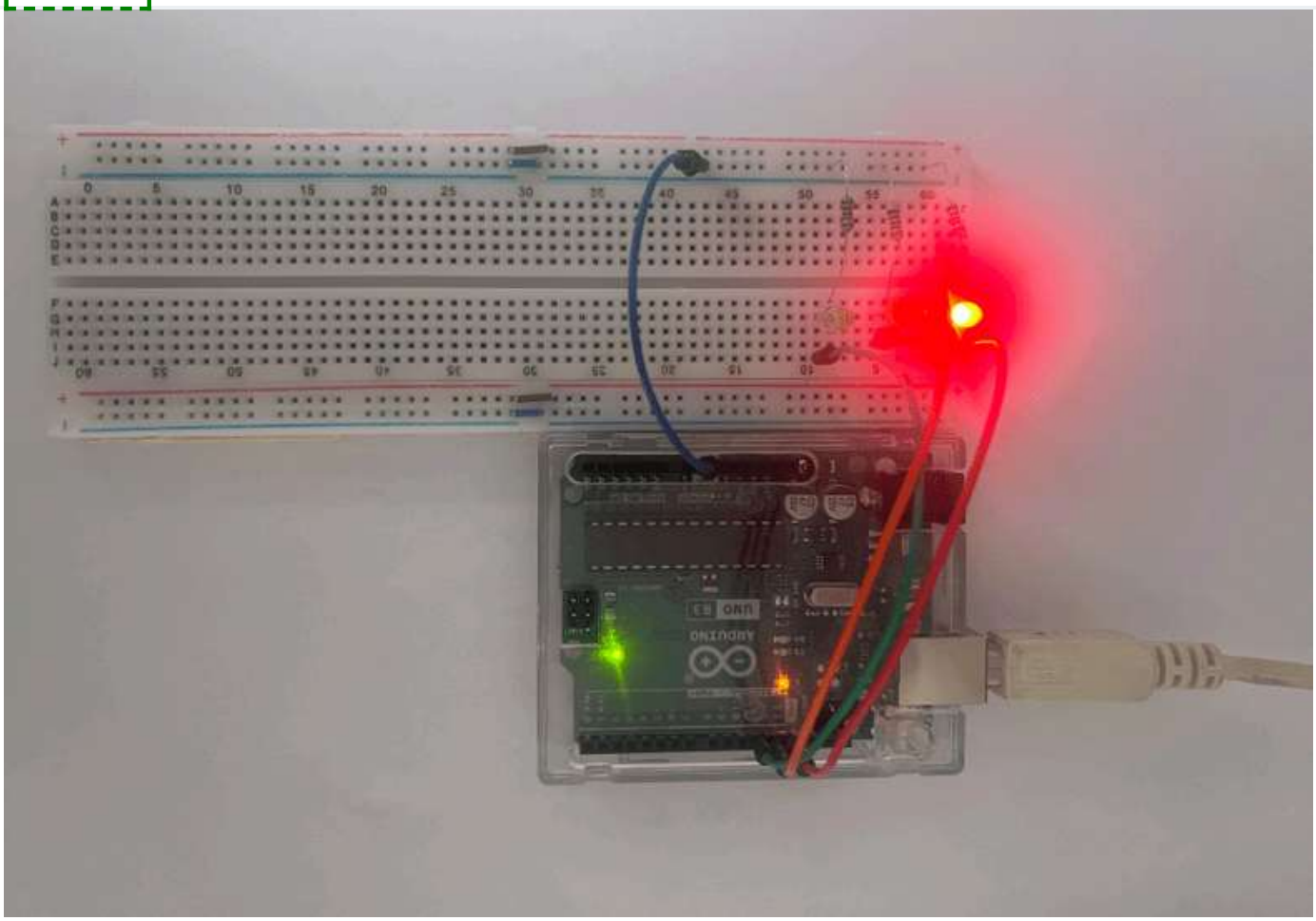
- La diode LED s'allume pendant 3 secondes. L'image ci-dessous nous le montre :



- La diode LED jaune s'alleume pendant 0.5 seconde. L'image ci-dessous nous le montre :



- La diode LED s'allume pendant 1 seconde. L'image ci-dessous nous le montre :



Fichier Arduino IDE (code source)

Vous pouvez télécharger le code source de cette activité au format projet **Arduino IDE** à partir de notre espace **GitHub** :

- **https://github.com/ElectroRobot/Arduino-feu-tricolore**